Reducing Text Bias in Synthetically Generated MCQAs for VLMs in Autonomous Driving

Anonymous ACL submission

Abstract

Multiple Choice Question Answering (MCQA) benchmarks are an established standard for measuring Vision Language Model (VLM) performance in driving tasks. However, we observe the known phenomenon that synthetically generated MCQAs are highly susceptible to *hidden textual cues* that allow models to exploit linguistic patterns rather than visual context. Our results show that a VLM fine-tuned on such data can achieve accuracy comparable to human-validated benchmarks even without visual input. Our proposed method reduces blind accuracy from +66.9% above random to +2.9%, eliminating the vast majority of exploitable textual shortcuts. By decoupling the correct answer from linguistic artifacts and employing a curriculum learning strategy, we force the model to rely on visual grounding, ensuring that performance accurately reflects perceptual understanding.

1 Introduction

Vision Language Models (VLMs) have become integral to the modern autonomous driving stack (NVIDIA et al., 2025; Hwang et al., 2025), especially for tackling safety-critical “long-tail” scenarios. While benchmarks fall into free-form QA (Park et al., 2025; Sima et al., 2023; Malla et al., 2023) or Multiple Choice Question Answering (MCQA) (Xie et al., 2025; Khalili and Smyth, 2025; Zeng et al., 2025; Lu et al., 2024), MCQA is often preferred due to its straightforward scoring and structured output.

However, the integrity of these benchmarks depends on the quality of distractors. Recent works have explored automated MCQA creation (Zhang et al., 2025a; Alhazmi et al., 2024; Loginova et al., 2025), but these methods do not guarantee the absence of *hidden textual cues*. Our investigation reveals that when VLMs generate both the correct answer and distractors for a driving task, they introduce subtle linguistic markers. We find that small VLMs trained on this data quickly learn to ignore visual inputs, achieving high accuracy by merely processing text. This is a form of shortcut learning that invalidates the benchmark as a measure of the trained VLM’s generalizability.

The core contributions to this work are:

- A framework for identifying hidden textual bias using trained VLMs with *zeroed-out visual inputs* during inference.
- A *two-stage MCQA generation* method, that first creates the question-answer pairs for each sample and second, samples distractors from answers of other samples, reducing linguistic bias towards the correct option.
- A *curriculum learning approach* that drops MCQ options early in training to encourage stronger visual-textual grounding.

2 Related Work

Foundation models for driving (Hwang et al., 2025) have gained popularity because they leverage internet-scale data to develop “physical common sense" and emergent reasoning abilities, allowing them to perform scene understanding by answering natural-language questions to handle the safety-critical “long-tail" scenarios that traditional modular systems struggle with. This usage encourages robust benchmarks to establish industry standards, validate safety in rare corner cases, and provide an objective measure of progress. While early benchmarks used graph-based VQA like DriveLM (Sima et al., 2023), modern frameworks like NuPlanQA (Park et al., 2025), DriveBench (Xie et al., 2025), DriveLM (Sima et al., 2023), and AutoDrive-QA (Khalili and Smyth, 2025) provide QA datasets at scale for supervised fine-tuning (SFT) and testing. These datasets however often lack an analysis of how synthetic MCQ options might introduce textual cues, allowing models to succeed without grounding their reasoning in the vision inputs.

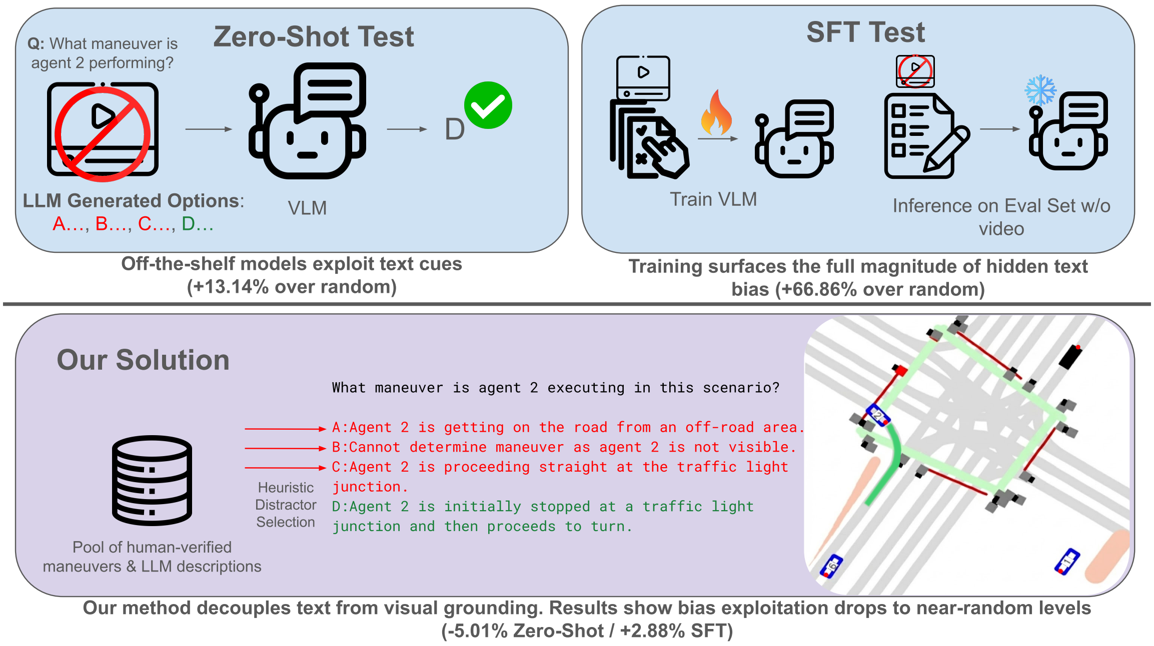

Figure 1: **Top**: We employ two video-disabled evaluations to detect textual cues. The Zero-Shot Test reveals inherent linguistic patterns in off-the-shelf models, while the SFT Test exposes the full magnitude of shortcut learning when a model is trained on biased data. **Bottom**: Our method replaces LLM-generated distractors with real descriptions sampled from elsewhere in the dataset. This reduces bias exploitation in both Zero-Shot and SFT.

The integrity of multiple-choice evaluation depends on the quality and plausibility of distractors. Modern agentic frameworks like AutoConverter (Zhang et al., 2025b), AutoDrive-QA (Khalili and Smyth, 2025), GOBBET (Lu et al., 2022) use language models to generate hard distractors. Many previous works have looked specifically into MCQA creation for LLM evaluation (Zhang et al., 2025a; Alhazmi et al., 2024; Loginova et al., 2025). While most of these focus on text only data, a recent work has shed light into hidden biases for VLMs (Loginova et al., 2025). In our work we find that there are more hidden textual cues that get exposed when we train a small model using this MCQ data. To address this, we propose a distractor selection workaround designed to decouple task-related signals from these textual artifacts.

3 Problem Setup and Synthetic MCQA Generation

In this section, we formalize synthetic MCQA construction for autonomous driving, defining the base driving dataset, the target MCQA format, and the data generation pipeline.

3.1 Driving Dataset Creation

We assume access to a labeled driving dataset $\mathcal{D}_{base} = \{(v_i, y_i)\}_{i=1}^{N}$, where v_i denotes a 10s birds-eye-view (BEV) video clip depicting a driving scene and $y_i \in \mathcal{Y}$ is the maneuver label describing the behavior of a selected non-ego agent in the scene. Maneuver labels are uniformly balanced and correspond to semantically meaningful agent behaviors such as lane changes, turns, etc. as defined in Appendix A or *agent not visible* when the selected agent is not in v_i.

From the base dataset $\mathcal{D}_{base}$, we construct a multiple-choice question answering (MCQA) task where each sample requires identifying the maneuver performed by a target agent in a driving scene. An MCQA instance is defined as $m_i = (q_i, v_i, \mathcal{A}_i)$, where q_i is a natural-language question referring to the target agent, v_i is the associated BEV video clip, and $\mathcal{A}_i$ is a set of answer options. Each answer set $\mathcal{A}_i$ contains one correct ground-truth answer a_i^{gt} and a set of plausible but visually incorrect distractors $\mathcal{A}_i^{cand} = \mathcal{A}_i \backslash a_i^{gt}$. Throughout our experiments we use a fixed four-way multiple-choice format.

Stage I: Natural Language Formatting of Ground-Truth Answer The first stage of synthetic MCQA construction produces a natural-language question–answer pair from maneuver label y_i for each sample. Given a base dataset entry (v_i, y_i), the question–answer pair is generated as $(q_i, a_i^{gt}) = f_{NL}(v_i, y_i)$, where f_{NL} is an expert model that maps structured maneuver labels to textual descriptions. The expert model (Gemini 2.5

Dataset	Model	Accuracy %
D_{llm}	Gemini 2.5 Pro	38.14 (+13.14)
D_{llm}	Gemini 2.5 flash	32.73 (+7.73)
D_{llm}	Qwen2-VL-7B	29.56 (+4.56)
D_{new}	Gemini 2.5 Pro	19.99 (-5.01)
D_{new}	Gemini 2.5 flash	19.88 (-5.12)
D_{new}	Qwen2-VL-7B	21.38 (-4.62)

Table 1: Results obtained by passing the MCQ through LLMs without video context. We present results as *Overall accuracy (Accuracy above random)*

for our experiments) is restricted from introducing semantic content beyond the ground-truth label.

Stage II (Baseline): LLM-Based Distractor Generation In the second stage of synthetic MCQA construction, we populate the answer set $\mathcal{A}_i$ by generating candidate distractors for each question-video-answer triple (q_i, v_i, a_i^{gt}) produced in Stage I. Distractors are generated using the same expert model employed during answer realization.

Formally, the answer set is constructed as $\mathcal{A}_i = \{a_i^{gt}\} \cup \mathcal{A}_i^{cand}$, where the candidate distractors are sampled as $\mathcal{A}_i^{cand} \sim p_{\text{LLM}}(\cdot \mid q_i, v_i, a_i^{gt})$. The resulting distractors are intended to be semantically plausible maneuver descriptions that do not match the visual content of v_i. Applying this two-stage procedure across the base dataset yields a synthetic MCQA dataset $\mathcal{D}_{\text{llm}} = \{(q_i, v_i, \mathcal{A}_i)\}_{i=1}^{N}$.

3.2 Empirical Analysis of Synthetic MCQs

We perform a series of analyses to assess whether the synthetically generated dataset D_{llm} exhibits unintended textual biases that allow models to perform well without visual grounding. Our analysis evaluates both off-the-shelf VLMs and models trained directly on the synthetic MCQs.

Zero-Shot Test We conduct a set of diagnostic tests designed to detect superficial cues in the question or answer options. We verify that the position of the correct answer within the answer set is uniformly distributed and manually inspect a random subset of MCQs to ensure there are no obvious textual giveaways (such as distractor lengths or jargon). We then evaluate multiple pretrained VLMs on D_{llm} using only textual inputs $(q_i, \mathcal{A}_i)$ while withholding the video v_i. As shown in Table 1, several models achieve accuracy above random chance in this setting, with the model used for data generation exhibiting the strongest performance. This suggests the presence of model-specific linguistic regularities embedded in the synthetic distractors.

Supervised Fine-tuned Test To uncover biases that emerge during training, we fine-tune a Qwen2-VL-2B model (Wang et al., 2024) on D_{llm} and evaluate it with shuffled answer ordering (Zeng et al., 2025) and zeroed-out visual inputs. For the zeroed-video setting, we partition the test set into D_{NV} (agent not visible is correct), D_N (agent not visible is an incorrect option), and D_V (agent not visible option absent). This split isolates distinct failure modes: D_{NV} tests whether the model can identify agent absence, while D_N and D_V probe reliance on textual cues when no visual evidence is available. Despite high nominal accuracy after fine-tuning (93.82%), performance remains high without video input; in particular, accuracy on D_V and D_N reaches 91.86% and 64.66% respectively (Table 2), indicating that the model recovers the correct answer based on the options' linguistic structure rather than a semantic mismatch between the distractor and the multimodal inputs.

4 Methodology

To eliminate text bias introduced during distractor construction, we modify the Stage II of synthetic MCQA generation by decoupling distractor selection from VLMs. In the baseline dataset D_{llm}, candidate answers $\mathcal{A}_i^{\text{cand}}$ are generated by an LLM conditioned on the ground-truth answer a_i^{gt}. In contrast, our proposed approach samples distractors in maneuver label space and reuses ground-truth answers from other samples.

After Stage I, each MCQA instance m_i is associated with a ground-truth maneuver label $y_i \in \mathcal{Y}$. For a given MCQA instance m_i, we construct the candidate answer set by sampling $K - 1$ distractor labels $\{y_j\}$ such that $y_j \in \mathcal{Y} \setminus \{y_i\}$. For each distractor label y_j we sample a ground-truth answer $a_k{}^{\text{gt}}$ such that $y_k = y_j$. To ensure contextual consistency, agent identifiers in these answers are rewritten to match the target agent referenced in the question. Applying this heuristic across the dataset yields a debiased MCQA dataset D_{new}, in which ground-truth answers and distractors are no longer coupled through a shared generative model.

Curriculum-Based Option Dropping In addition to modifying distractor construction, we employ a curriculum-based training strategy to further discourage reliance on textual cues. During supervised fine-tuning, answer options are randomly dropped from a fraction of training samples, converting MCQA instances $(q_i, v_i, \mathcal{A}_i)$ into open-

Dataset	Training Strategy	Accuracy (%) ↑		Accuracy Video Zeroed Out (%)		
		Normal	Shuffled	(D_{NV}) ↑	(D_N) ↓	(D_V) ↓
D_{llm}	No Training	29.23%	6.46%	17.48%	32.22%	23.4% (-1.6%)
D_{llm}	Regular SFT	93.82%	89.03%	99.68%	64.66%	91.9% (+66.9%)
D_{new}	No Training	24.38%	6.23%	12.94%	29.85%	23.1% (-1.9%)
D_{new}	Regular SFT	75.72%	66.43%	**100%**	0.21%	**27.9% (+2.9%)**
D_{new}	Curriculum	77.28%	70.82%	**100%**	0.84%	30.4% (+5.4%)
D_{new}	Regular SFT (FF)	88.08%	84.35%	**100%**	0.21%	29.5% (+4.5%)
D_{new}	Curriculum (FF)	**92.09%**	**87.64%**	**100%**	**0.0%**	29.9% (+4.9%)

Table 2: Qwen2-VL-2B training and evaluation results on different datasets and training strategies. Models are trained with ViT and projector frozen unless noted as full fine-tune *(FF)*. Model evaluation uses greedy sampling ($k = 1$) to eliminate variance. For D_V we present results as *Overall accuracy (Accuracy above random)*.

ended question (q_i, v_i) with target output answer a_i^{gt}. This forces the model to generate the correct answer conditioned on visual input rather than selecting from a fixed option set $\mathcal{A}_i$.

We schedule the fraction of option dropping $x(t)$ for training step t according to

$$x(t) = \max\Big(d_{\min},\, d_{\max} - d_{\min}\,(t/\tau)^2\Big), \quad (1)$$

where τ is the number of training steps over which the curriculum is applied, and $d_{\min}$ and $d_{\max}$ denote the minimum and maximum drop ratios, respectively. Early in training, options are dropped aggressively to encourage visual grounding, and are gradually reintroduced as training progresses.

5 Results & Discussion

We evaluate whether the debiased dataset D_{new} eliminates textual shortcuts under zero-shot and supervised fine-tuning (SFT) settings. Unlike the baseline D_{llm}, models evaluated on D_{new} perform near chance without visual input.

Zero-shot evaluation We first assess textual bias using pretrained VLMs evaluated without video input. Models evaluated on D_{new} achieve accuracy close to random guessing (-5.01%), in contrast to D_{llm} (+13.14%), where several models perform significantly above random chance (Table 1). Notably, the model used to generate the baseline dataset exhibits the strongest bias on D_{llm}, while its performance drops below random on D_{new}. This confirms that heuristic distractor sampling in maneuver label space removes model-specific linguistic cues that are exploitable in zero-shot settings.

Supervised fine-tuning evaluation We next evaluate whether textual bias re-emerges after training by fine-tuning a Qwen2-VL-2B model under controlled conditions. When trained on D_{llm}, the model achieves high accuracy (93.82%) and retains near-human performance even when the video is zeroed out (+66.86%). In contrast, models trained on D_{new} exhibit substantially reduced performance without visual input (Table 2), particularly on the D_V subset, where accuracy remains close to random (+2.88% for regular SFT). The performance on D_N subset being close to 0% (+0.21% for regular SFT) and D_{NV} subset being 100% also indicate that the model can no longer recover the correct answer from textual patterns alone and must rely on visual evidence to succeed.

Effect of curriculum-based training We further observe that curriculum-based training improves robustness. Compared to regular SFT, curriculum training yields higher accuracy while reducing sensitivity to answer shuffling and video removal. This suggests improved visual-textual grounding rather than increased reliance on option structure. The strongest performance is achieved when curriculum learning is combined with full fine-tuning of the vision encoder and projector, reflecting the importance of adapting visual representations for BEV inputs, which are uncommon in standard VLM pretraining. High performance on D_{new} therefore reflects genuine visual understanding rather than linguistic bias, validating the effectiveness of the proposed debiasing approach.

6 Conclusion

In this work, we systematically demonstrate that synthetically generated MCQAs are subject to hidden textual cues introduced by the VLMs used to generate them. We propose a distractor generation method that reduces this bias and a curriculum learning modification to improve grounding when using such data in training mixes. Our results confirm that these methods force models to rely on visual information rather than linguistic shortcuts.

Limitations

We limit our analysis in this paper to training the Qwen2-VL-2B model for the maneuver classification task due to resource constraints. Training on more datasets and models can provide a deeper understanding about the hidden patterns and the conditions where they are more evident. Investigating similarity scoring like BLEU, word frequencies or embedding space distance for MCQAs can also uncover biases in the Zero-Shot test before subjecting models to VLM training. In our work we keep the zero-shot tests simple while leaving the majority of bias detection to model training as our main goal was to train a model to answer driving related questions in addition to developing a reliable benchmark.

In our baseline experiments we use BEV videos with heuristic maneuver labels to generate MCQAs. The expert models might have limited exposure to such data resulting in the text patterns. Reproducing the experiments with a more common video classification tasks that the expert model has exposure to can give a more complete insight.

References

Elaf Alhazmi, Quan Z. Sheng, Wei Emma Zhang, Munazza Zaib, and Ahoud Alhazmi. 2024. Distractor generation in multiple-choice tasks: A survey of methods, datasets, and evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14437–14458, Miami, Florida, USA. Association for Computational Linguistics.

Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. 2025. Emma: End-to-end multimodal model for autonomous driving. *Preprint*, arXiv:2410.23262.

Boshra Khalili and Andrew W. Smyth. 2025. Autodrive-qa: A multiple-choice benchmark for vision-language evaluation in urban autonomous driving. *Preprint*, arXiv:2503.15778.

Olga Loginova, Oleksandr Bezrukov, Ravi Shekhar, and Alexey Kravets. 2025. Addressing blind guessing: Calibration of selection bias in multiple-choice question answering by video language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3216–3246, Vienna, Austria. Association for Computational Linguistics.

Jiaying Lu, Xin Ye, Yi Ren, and Yezhou Yang. 2022. Good, better, best: Textual distractors generation for multiple-choice visual question answering via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4921–4930.

Yuhang Lu, Yichen Yao, Jiadong Tu, Jiangnan Shao, Yuexin Ma, and Xinge Zhu. 2024. Can lvlms obtain a driver's license? a benchmark towards reliable agi for autonomous driving. *Preprint*, arXiv:2409.02914.

Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. 2023. Drama: Joint risk localization and captioning in driving. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1052.

NVIDIA, :, Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, Liang Feng, Greg Heinrich, Jack Huang, Peter Karkus, Boyi Li, Pinyi Li, Tsung-Yi Lin, and 25 others. 2025. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *Preprint*, arXiv:2511.00088.

Sung-Yeon Park, Can Cui, Yunsheng Ma, Ahmadreza Moradipari, Rohit Gupta, Kyungtae Han, and Ziran Wang. 2025. Nuplanqa: A large-scale dataset and benchmark for multi-view driving scene understanding in multi-modal large language models. *arXiv preprint arXiv:2503.12772*.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. 2023. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Shaoyuan Xie, Lingdong Kong, Yuhao Dong, Chonghao Sima, Wenwei Zhang, Qi Alfred Chen, Ziwei Liu, and Liang Pan. 2025. Are vlms ready for autonomous driving? an empirical study from the reliability, data and metric perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6585–6597.

Tong Zeng, Longfeng Wu, Liang Shi, Dawei Zhou, and Feng Guo. 2025. Are vision llms road-ready? a comprehensive benchmark for safety-critical driving video understanding. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 5972–5983, New York, NY, USA. Association for Computing Machinery.

Yuhui Zhang, Yuchang Su, Yiming Liu, Xiaohan Wang, James Burgess, Elaine Sui, Chenyu Wang, Josiah

Aklilu, Alejandro Lozano, Anjiang Wei, Ludwig Schmidt, and Serena Yeung-Levy. 2025a. Automated generation of challenging multiple-choice questions for vision language model evaluation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29580–29590.

Yuhui Zhang, Yuchang Su, Yiming Liu, Xiaohan Wang, James Burgess, Elaine Sui, Chenyu Wang, Josiah Aklilu, Alejandro Lozano, Anjiang Wei, Ludwig Schmidt, and Serena Yeung-Levy. 2025b. Automated generation of challenging multiple-choice questions for vision language model evaluation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

A Appendix

A.1 Maneuver Types

The maneuver determination for selected agents is done using heuristic conditions on their longitudinal and lateral movement on the road in a given scene. The maneuver types used for data generation and grouping for distractor section are as follows:

1. **STRAIGHT** - The agent is going straight
2. **TURNING** - The agent is making a turn in the junction
3. **NUDGE AROUND OBSTACLE** - The agent is going around an obstacle like Vehicle, pedestrian, garbage etc..
4. **LANE CHANGE** - The agent is making a lane change
5. **REVERSING** - The agent is moving backwards on the road
6. **U TURN** - The agent is making a U turn
7. **GETTING ON ROAD** - The agent is transitioning from being off road to being on road
8. **PARKING LANE CUTIN** - The agent is transitioning from being in the parking lane to being in the driving lane
9. **STATIONARY** - The agent is parked or double parked
10. **GOING OFF ROAD** - The agent is transitioning from being on road to being off road
11. **STOPPED** - The agent is stopped for traffic light or stop sign

A.2 Dataset Details

We use the same samples D_{base} and Stage I processing for D_{llm} and D_{new}. The driving dataset D_{base} contains 59940 samples, out of which, we hold out 1796 samples for the test set. We hold out the same samples for D_{llm} and D_{new} to keep the accuracy numbers comparable. We begin with base dataset D_{base} containing close to uniform label distribution for the 12 maneuver types resulting in 7395 (8.33%) samples per maneuver type and we maintain the uniformity for the holdout set as well by sampling 150 or 149 samples per maneuver type. We run an automated pipeline with the BEV videos to detect when the agent in question is not visible in it and replace the base maneuver label with *agent not visible* label. This results in forming the final D_{base} that is used for data generation, with 18.8% of the data being marked as *agent not visible* while keeping rest of the maneuver labels close to uniform, ranging between 6% to 8% of the data distribution as shown in Table 3. We also ensure that the label distribution is close to uniform in our train and test datasets to avoid model bias as seen in Figure 2.

Maneuver Type	**Original (%)**	**Processed (%)**
Not Visible	—	18.82
Reversing	8.34	7.89
Turning	8.34	7.13
Nudge Around	8.32	7.13
Hard Stopped	8.33	6.88
Going Off Road	8.34	6.82
Getting On Road	8.32	6.68
U-Turn	8.34	6.61
Straight	8.34	6.60
Parking Cut-in	8.33	6.59
Lane Change	8.34	6.45
Stopped	8.33	6.39
Stationary	8.34	6.00

Table 3: Maneuver type distribution of D_{base} showing original uniform tags vs processed tags with visibility logic applied.

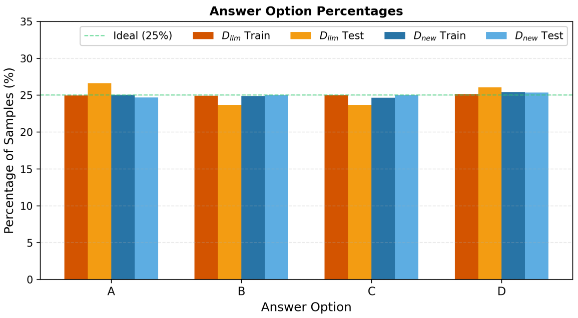

Figure 2: Distribution of the correct option in the train and test subsets of D_{llm} and D_{new}. We see even distribution of correct answers for all datasets.

A.3 Manual Review

A team of human expert labelers manually reviewed 360 randomly sampled MCQs from the test set of D_{llm} after Stage II of data generation. Their answers are compared against the ground truth to obtain an accuracy score. These accuracy scores act as a baseline human evaluation score used for comparing model performance. The average accuracy among the expert labelers was 88.7% with an inter-labeler agreement of 93.9%. This demonstrates that the questions may be consistently answered between reviewers and provides an approximate baseline of accuracy numbers we might expect from fully-trained models. Models that outperform the human expert baseline may be doing so for a variety of reasons, such as a small sample size used to collect the baseline, input cueing, or learning patterns on incorrect labels.

A.4 Results on Pretrained VLMs

In addition to calculating accuracy of the pretrained models on the two datasets in Table 1, we also run shuffle evaluation without video. When we shuffle evaluate without video on pretrained models, we ideally expect close to 0 accuracy as the chance of picking the wrong answer in one of the four variants is high. We observe that D_{llm} has higher performance suggesting more bias compared to D_{new}. Similar to the accuracy numbers in Table 1, Gemini 2.5 Pro which was used to create the data exploits the biases most. Additionally these biases again do not expose the complete extent of bias in the text as seen with shuffle results in Table 2.

Dataset	Model	Shuffle
D_{llm}	Gemini 2.5 Pro	19.71
D_{llm}	Gemini 2.5 flash	9.79
D_{llm}	Qwen2-VL-7B	13.86
D_{new}	Gemini 2.5 Pro	8.68
D_{new}	Gemini 2.5 flash	5.57
D_{new}	Qwen2-VL-7B	6.40

Table 4: Results obtained by passing the MCQ through LLMs without video context.

A.5 Hyperparameters for training and inference

The hyperparameters we used for our training runs are shown in Table 5. We maintain consistent settings across all training runs, employing early stopping once the validation loss begins to increase. Extended training runs confirm the absence of the double descent phenomenon, justifying our use of early stopping at the first inflection point. Curriculum-specific configurations are applied only for curriculum learning runs. For evaluations we use greedy sampling, i.e. $k = 1$ whenever possible. We train a Qwen2-VL 2B parameter model for 20hrs on 8 B200 GPUs per training run.

Hyperparameter	Value
Model Configuration	
Base Model	Qwen2-VL-2B
Precision	bfloat16
Optimization	
Optimizer	AdamW
Learning Rate	2×10^{-5}
Weight Decay	0.1
Decay Type	Linear
Warmup steps	100
Training Setup	
Global Batch Size	256
Random Seed	42
Max steps	2500
Curriculum Learning	
d_{min}	0
d_{max}	100
τ	670

Table 5: Hyperparameters for Qwen2-VL-2B Maneuver MCQ Fine-tuning. The curriculum learning parameter τ refers to the number of training steps over which curriculum is applied, d_{min} and d_{max} refer to the minimum and maximum drop rations.